# A Probabilistic Approach for Reducing the Search Cost in Binary Decision Trees

*A. Rontogiannis & N. J. Dimopoulos*

Dept of Electrical & Computer Engineering
University of Victoria
P.O. Box 3055, Victoria, B.C., V8W 3P6 CANADA

*Abstract*-**In this work, we present a probabilistic model for reducing the number of decisions (tests) that are required in a particular diagnostic procedure. Specifically, we consider that a problem is structured as a binary balanced decision tree the interior nodes of which represent test points; the paths of the tree correspond to different diagnoses. By assuming that there exists sufficient probabilistic information available concerning the decisions at the interior nodes, we attempt to minimize the average number of these decisions when we search for a final diagnosis.**

## I. INTRODUCTION

People often cope with problems in which they have to make decisions under conditions of uncertainty. This is the case because some components of the problem under consideration are only partially known or the cost associated with obtaining all the necessary information is high. When the decisionmaker faces such a situation he must base his decisions and conclusions on his experience or on available information. The whole procedure can be facilitated when it is properly structured in a way that clarifies the particularities of the problem. Decision trees constitute an effective representation for modeling the human reasoning. Different paths of the decision tree express different policies that can be followed in order for a solution to be found. Probability theory and statistics provides a range of tools by means of which predictions can be made based on the assumption that future events will fit the predefined model.

The decision tree structure can be used either for maximizing an expected utility function or for expressing classification rules. In the former case ([1],[2],[3]), the interior nodes of the tree are divided into *decision nodes* where the decisionmaker is in control of the choice and *chance nodes* which correspond to events (often called "states of nature") that lie outside the control of the human expert. The main goal of the decisionmaker is to follow the path which maximizes an expected utility function that is related to the problem parameters. In [4],[5],[6] and [7] the decision tree structure is used as a classification tool. The leaves of the tree constitute a disjoint set of classes while at each interior node the value of a particular attribute is checked. In [4], the author describes how such a tree can be constructed from a set of initial objects that are distinguished through the values of different attributes. Then, that tree can be used for the classification of new objects.

In our work a different appoach to decision problems is adopted. Specifically, we consider binary balanced decision trees and assume (as in [7]) that there exist sufficient probabilistic (statistical) information available concerning the decisions at the interior nodes. Our basic goal is to develop a model in which, relying on the given information, we reduce the number of decisions that are required when we search for a conclusion. That can be achieved by selecting the nodes at which an explicit decision is not taken but instead the branch of the tree that is to be followed is determined probabilistically. Such an approach has the effect that we may sometimes end up with wrong conclusions. We can then search for the correct path by appropriately backtracking into the tree. We must therefore choose both the nodes where we will not test for the value but we shall attain the highest probability value, and the proper backtracking method in case that a wrong conclusion is reached so that the average number of tests performed for every single conclusion is minimized. We have formulated a gain function and we present a number of decision-tree traversing procedures which maximize this gain.

## II. THE GAIN FUNCTION

In order to develop our mathematical model and create the gain function that is subject to maximization we make the following assumptions

- We consider binary balanced trees of depth $n$.
- All the tests contained in the interior nodes are of the same significance.
- We know the probabilities of all the tests' outcomes. Thus, each interior node holds two probabilities which express the frequency of occurence of the two possible outcomes of the corresponding test.
- The leaves of the tree constitute a set of distinct final conclusions.
- When we end up with a wrong path we can backtrack and search for the correct one.
- We are able to distinguish between correct and wrong final outcomes without any cost.

Before we proceed to the expression of the gain function we give the following definition of a *statistical node*

***Definition 1.1.*** *An interior node of the tree is called statistical when the test is not performed at that node but instead the decision is based on the probabilistic information that we have. The branch of the node which is followed is called a statistical branch.*

Each statistical node has one statistical and one non-

 0-7803-0971-5/93/$3.00 © 1993 IEEE

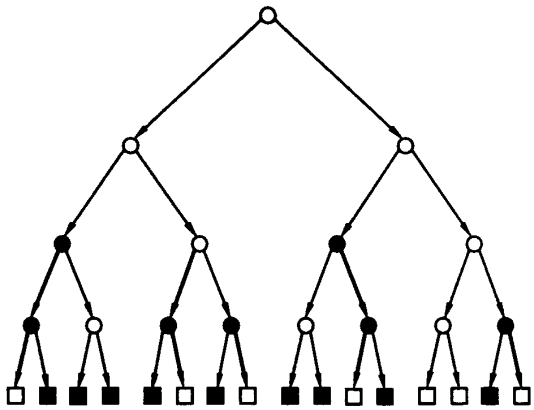

***Figure 1.*** Example Tree

statistical branch. Each non-statistical node has two non-statistical brances. By partitioning the interior nodes of the tree into statistical and non-statistical, we separate the set of final nodes $F$ into two disjoint subsets : the subset of the reachable nodes $F_1$ and the subset of the non-reachable nodes $F_2$. This can become more obvious by considering the tree of figure 1. The tree has depth $n = 4$. The shaded circles represent the statistical nodes while the statistical branches are highlighted. The shaded squares constitute the non-reachable final nodes. In general, a path which corresponds to a reachable final node has the following characteristics

1. The number $m_j$ of its statistical nodes.
2. The probability $P_{Pj}$ that the path is going to be followed. $P_{Pj}$ can be expressed as the product of the probabilities of the non-statistical branches of the path ([8]).
3. The probability $P_{Cj}$ that the path is correct given that it has been followed. $P_{Cj}$ is the product of the probabilities of the statistical branches of the path ([8]).
4. The backtracking cost $B_j$. The backtracking cost represents the number of tests that take place after the path $j$ turns out wrong and before the correct path is found. Its general expression is given in the next section.

We can show that the gain $G$ which expresses the number of tests that are saved in a single traversal of the tree when the probabilistic information is taken into account can be written as follows

$$G = \sum_{j \in F_1} P_{Pj} \cdot [m_j - B_j \cdot (1 - P_{Cj})] \qquad (1)$$

For a particular partition of the interior nodes of the tree, all the components of equation (1) except for $B_j$ are known and well-defined. Therefore, in order for the gain $G$ to be maximized, two specific tasks must be accomplished: a) the optimal partition of the nodes must be found and b) the backtracking cost $B_j$ for each reachable final node resulting from that partition must be minimized. The general expression of $B_j$ is given in the next section. In section IV two partitioning techniques are proposed. The problem of finding the optimal partition appears to be a difficult and complicated one.

## III. The Backtracking Cost

As it was stated in the previous section, when we end up with a wrong diagnosis we backtrack into the tree and search for the correct path. Such a procedure results in a cost which is called backtracking cost. The backtracking cost represents the number of tests that take place after a wrong path is diagnosed and before the correct one is found. In order to simplify our model we assume that during the search for the correct path, after backtracking, the probabilistic information that is available at the nodes of the tree is not taken into account any longer.

From (Eq. 1) it is clear that for a specific partitioning of the interior nodes the gain $G$ is maximized if the backtracking cost $B_j$ is minimized for every reachable node $j$. In this section a single path of the tree is considered and the general expression of its backtracking cost is given. The results can be applied to each path of the set $F_1$.

If we follow a path which has $m$ statistical nodes, say $S_1, S_2, \ldots, S_m$, and the final diagnosis turns out wrong, that happens because in at least one statistical node a wrong decision was taken. Therefore, in the backtracking procedure, only the statistical nodes need to be considered. A particular statistical node $S_i$, $1 \le i \le m$, has two characteristics

- The *height* $h_i$, which also represents the number of tests between that node and a leaf node.
- The *probability of correct backtracking* $P_i$. $P_i$ is the probability that the first wrong decision took place either at the node $S_i$ or at a node which follows $S_i$ in the path $(S_{i+1}, \ldots, S_m)$. In other words $P_i$ expresses the probability that we are going to find the correct final node after backtracking to the node $S_i$ and crossing the tree by performing all the tests at the nodes we meet. $P_i$ can be calculated exclusively from the probabilities of the statistical nodes of the path ([8]).

The general expression of the backtracking cost for a single path of the tree is derived as

***Theorem 2.1.*** *If $S_{N_1}, S_{N_2}, \ldots, S_{N_q}$ are the $q$ statistical nodes at which we back up with $1 \le q \le m$, $S_{N_q}$ the first node at which we backtrack and $S_{N_1} \equiv S_1$, the backtracking cost $B$ can be written in the following form*

$$B = \sum_{i=1}^{q} h_{N_i} \cdot (1 - P_{N_{i+1}}), \quad P_{N_{q+1}} = 0 \qquad (2)$$

To achieve the minimum value of the backtracking cost $B$ we solve the following optimization problem

$$min\left[\sum_{i=1}^{q} h_{N_i} \cdot (1 - P_{N_{i+1}})\right] \qquad (3)$$

In general, if there are $m$ statistical nodes, we have $2^{m-1}$ different backtracking procedures which result in $2^{m-1}$ costs. That is so because the last node at which we back up is always node $S_1$ and $2^{m-1}$ is the number of all the possible combinations of the remaining $m-1$ nodes. Out of these $2^{m-1}$ procedures there is one (or probably more) with the minimum cost given in (3). In other words there is a combinations of $q$ nodes, $1 \le q \le m$, which minimizes the backtracking cost given in (2). By exhaustively examining all the possible combinations, the minimum cost $B$ for a particular path of the tree can be obtained. If we then repeat the same procedure for every path $j$ of the set $F_1$ and substitute the results in (1) the maximum $G$ for a specific partition of the interior nodes of the tree can be achieved. We must note that it is not necessary to calculate the costs of all the possible backtracking procedures. The backtracking process which results in the minimum cost has special properties the application of which reduces significantly the number of cost evaluations. These properties are summarized in the following (for the proofs see [8]):

***Proposition 1.*** The optimal backtracking procedure, that is, the one which achieves the minimum cost given in (3), does not contain a node $S_l$ such that $h_l/p_l > h_1$ where $h_1$ is the height of node $S_1$

***Proposition 2.*** The optimal backtracking procedure does not contain two consecutive nodes $S_l$ and $S_k$ such that $h_k < h_l/P_l$

***Proposition 3.*** The optimal backtracking procedure does not start at a node which is located below the node with the minimum value of $h_i/P_i$, $1 \le i \le m$.

## IV. Partitioning of the Interior Nodes

It was mentioned before that in order for the maximum value of the gain $G$, given in (1), to be attained the appropriate partition of the interior nodes of the tree must be found. This partition is called optimal. Under the hypothesis that the optimal separation of the nodes is given, the maximum gain $G$ can be obtained by minimizing the backtracking cost $B_j$ for all $j$ in $F_1$. In general, if there exist $s$ interior nodes, $2^s$ possible partitions of them into statistical and non-statistical can be encountered. For each of them the maximum gain must be calculated according to (1) and (3) and the results must be compared to each other in order for the optimal gain to be obtained. More specifically, in a tree of depth $n$ there exist $2^n - 1$ interior nodes. As it is clear the corresponding number of possible partitions is large . Even for a tree with $n = 5$ that number is too high ($2^{31}$) resulting in an excessive amount of computations during the maximization procedure. For these reasons the exhaustive evaluation of the gain for each possible partition is avoided. However, in the following two partitioning techniques are proposed. In both of them while the computational complexity is kept relatively low, the gain obtained seems to be high enough.

### *A. The Threshold Method*

As it was stated in section 1, each interior node (test) $k$ holds two probabilities that add up to 1 and represent the frequency of occurence of the two possible test outcomes. Obviously, the maximum of these two probabilities, say $P^k_{max}$ , is greater than or equal to 0.5. We define a *threshold t* to be a number in the interval [0.5, 1]. When the node $k$ is reached $P^k_{max}$ is compared to the threshold $t$. If it is greater than or equal to $t$ we follow the branch with the maximum probability without performing any test at the node. Otherwise, the branch which will be followed is determined according to the test outcome. Any threshold $t$, $0.5 \le t \le 1$, separates the interior nodes of the tree into statistical and non-statistical as follows : all the interior nodes the maximum probability of which is greater than or equal to $t$ are considered statistical and the rest become non-statistical. Therefore , the gain $G$ is expressed as a function of the probability threshold $t$ in the interval $[0.5, 1]$.

In a binary tree of depth $n$ there are $s = 2^n - 1$ interior nodes and therefore there exist $s$ probabilities that are greater than or equal to 0.5. For simplicity and without loss of generality we can assume that these probabilities are distinct and different than 0.5 and 1. Under this assumption, these probabilities divide the interval [0.5, 1] into $2^n$ subintervals. Among the subintervals there is one (or probably more), say $(p_r, p_{r+1}]$ , such that if $t \in (p_r, p_{r+1}]$ the gain $G$ is maximized. The value of $G$ is constant in each of the subintervals. That is so because, for each value of the threshold in a particular interval the resulting partition of the nodes is the same. In order to get the maximum value of $G$, (Eq. 1) must be evaluated in each of the subintervals. We must note that the maximum of $G$ so obtained is not the absolute maximum discussed before. It simply is the optimal gain achieved by applying the threshold method. The gain as a function of the threshold for a tree of depth 4 is shown in figure 2. For the production of the probabilities of the tree a uniform random generator has been used.

### *B. An Alternative Method*

Irrespective of their heights, the threshold method handles nodes with the same maximum probability in like manner. But, in general, a node which is located in ligher levels of the tree corresponds to a higher backtracking cost. Therefore, except for the the maximum probability $P^k_{max}$, the height of an arbitrary node $k$ must be also taken into account when we search for a partition of the interior nodes of the tree.

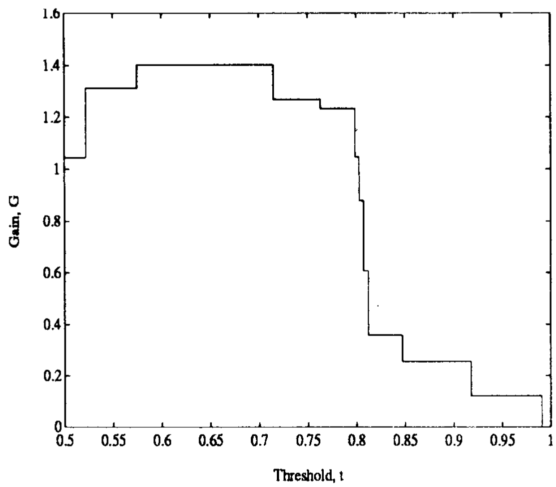


***Figure 2.*** The Gain versus the Threshold

We observe from equation (2) that the backtracking cost associated with a single path is increased when statistical decisions take place at high levels of the tree. That happens because when we end with a wrong path we back up with some probability to high levels of the tree and then we have to cross the tree by performing all the tests at the nodes we meet. Therefore, if we reduce the number of probabilistic decisions in such nodes and at the same time we increase the number of probabilistic decisions in nodes with relatively small height, we may obtain a higher value for the gain $G$. In the new technique, a different partition of the interior nodes of the tree results. In order to keep concistency with the previous method we assume, even though it is not necessary, that in the statistical nodes, the branch with the maximum probability is still followed.

In the new method we make a breadth-first visit of all the interior nodes of the tree starting from the last level. At each node we calculate the total gain of the paths which contain that node by considering it as a statistical one. The gain so obtained, is compared with the sum of the gains of the two subtrees originating at the node and if is greater than that, the node becomes statistical. Otherwise it remains non-statistical and we proceed to the next node of the same level of the tree (if any) or to the first node of the next level. We note that the gains of the two subtrees have already been calculated in the previous level of the tree.

The performance of the two methods is illustrated in table 3. In this table we consider trees of depth 8. In both cases a uniform random generator is used for the production of the probabilities of the branches. More specifically, the probabilities of the left branches of each node are uniformly distributed in the interval [0, 1]. Six different uniform distributions are considered in table 1. $G_1$ and $G_2$ stand for the gain in the threshold and the alternative method respectively. The second and the fourth column keep the corresponding percentages of saved tests. We observe that in general the second method performs better than the threshold method.

| $i$ | $G_1$ | % | $G_2$ | % |
|---|---|---|---|---|
| 1 | 1.907 | 23.8 | 2.284 | 28.6 |
| 2 | 1.873 | 23.4 | 2.228 | 27.9 |
| 3 | 1.719 | 21.5 | 2.195 | 27.4 |
| 4 | 1.107 | 14.8 | 1.639 | 20.5 |
| 5 | 1.960 | 24.5 | 2.479 | 31.0 |
| 6 | 1.271 | 15.9 | 2.016 | 25.2 |

***Table 1.*** Performance of the two Methods for $n = 8$

## V. SUMMARY-CONCLUSIONS

In this paper we presented a probabilistic model for reducing the number of tests that are required is a specific decision procedure. We assumed that a problem is structured as a binary balanced decision tree and we attempted to select the nodes of the tree where a probabilistic decision is taken. A gain function was built up and the expression of its parameters was derived. Two heuristic methods were proposed for the selection of the nodes were a decision is taken probabilistically and they were compared to each other in terms of the value of the gain achieved. Our model can be probably improved by investigating other methods for the selection of the nodes of the tree.


## ACKNOWLEDGEMENT

This work was supported under a grant from the Canadian Cable Labs Fund.


## REFERENCES


[1] R. E. Neapolitan, *Probabilistic Reasoning in Expert Systems*, John Wiley & Sons Inc., pp. 371-380, 1990.

[2] M. Hamburg, *Statistical Analysis of Decision Making*, Harcourt, Brace & World Inc., pp. 726-737, 1970.

[3] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Mateo, California, 1988.

[4] J. R. Quinlan, "Decision Trees and Decisionmaking", *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 2, pp. 339-346, March/April 1990.

[5] C. Carter, J. Catlett, "Assessing Credit Card Appplications using Machine Learning", *IEEE Expert*, vol. 2, no. 3, pp. 71-79, Fall 1987.

[6] K. B. Irani, J. Cheng, U. M. Fayyad, Z. Qian, " Applying Machine Learning to Semiconductor Manufacturing", *IEEE Expert*, vol. 8, no. 1, pp. 41-47, February 1993.

[7] J. R. Quinlan, "Decision Trees as Probabilistic Classifiers", *Proc. Fourth. Int. Workshop on Machine Learning*, P. Langley, Ed. Los Altos, CA : Morgan Kaufmann, pp. 31-37, 1987.

[8] A. Rontogiannis, "A Probabilistic Approach for Reducing the Search Cost in Binary Decision Trees", *Master Thesis*, University of Victoria, 1993.